

Cátedra de Ingeniería de Software

Código Asignatura: 00825

Asignatura: Estructura de Datos

Cuatrimestre: Primer Cuatrimestre 2020

Por: Roberto Chacón Chaverri.

Tarea #1

Objetivo:

Que el estudiante utilice recursividad para resolver problemas comunes utilizando el lenguaje Java.

Explicación:

La recursividad es una técnica de programación que permite que un bloque de instrucciones se ejecute n veces. Java nos permite utilizar recursividad ya que en este lenguaje es posible que un método se invoque a sí mismo. Debe quedar claro que no se trata de iteración como la que utilizan otras estructuras como el for, while, etc.

Desarrollo:

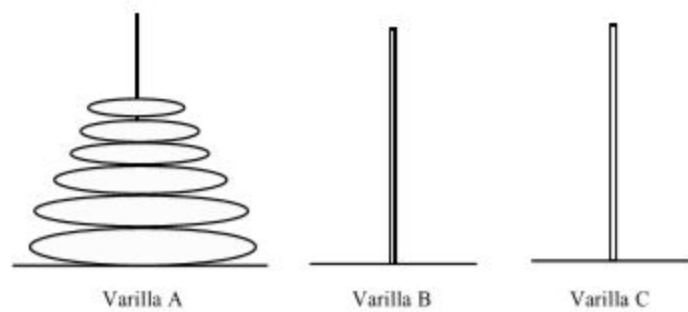
Las Torres de Hanoi es un algoritmo clásico que consta de tres columnas o varillas llamadas origen, destino y auxiliar y una serie de discos de distintos tamaños. Los discos están colocados de mayor a menor tamaño en la columna origen. La idea consiste en pasar todos los discos a la columna destino y dejarlos como estaban de mayor a menor (el más grande en la base, el más pequeño arriba).

Las reglas del juego son las siguientes:

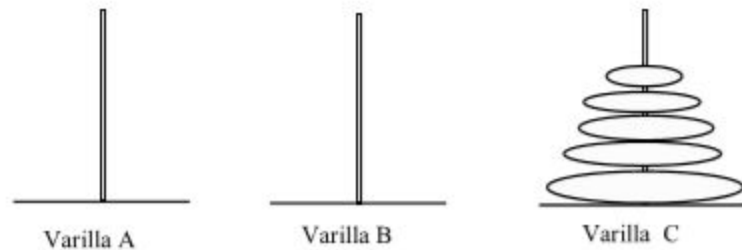
- Sólo se puede mover un disco cada vez.
- Para cambiar los discos de lugar se pueden usar las tres columnas.
- Nunca deberá quedar un disco grande sobre un disco pequeño.

Se debe crear una aplicación Java que utilice recursividad y que resuelva el problema de las Torres de Hanoi.

La posición inicial de las Torres es la siguiente:



La ejecución del programa finaliza cuando la posición de los discos sea como en la siguiente figura:



Además de la ejecución correcta del algoritmo, se debe mostrar en pantalla las diferentes posiciones que van tomando los discos (paso a paso) mientras se van trasladando de la Varilla A a la C, pasando por la Varilla B.

La forma de mostrar esto en pantalla debe ser gráfica, es decir, deben mostrarse los dibujos como los que se ven en el ejemplo de este enunciado.

Existirá un botón INICIAR, que iniciará la “animación” de los pasos que se van ejecutando hasta completar la solución, desde el paso 1 (con todos los discos en la varilla A) hasta el último paso (con todos los discos en la Varilla C).

Deben incluirse validaciones para que el programa nunca dé errores numéricos ni de otro tipo.

Rúbrica:

Aspecto a calificar	Puntos
Uso de recursividad	1
Correctitud del algoritmo de las Torres de Hanoi	1
Visualización gráfica de los movimientos de los discos en las columnas o varillas (como se ve en los dibujos de este enunciado)	1
El programa se ejecuta sin errores	1
TOTAL	4

Tarea #2

Objetivo:

Que el estudiante aplica el concepto de pila estática utilizando Java.

Explicación:

Una pila es una estructura de datos donde el último elemento en ingresar es el primero en salir (LIFO: Last in, First out). Por ejemplo, podemos mencionar una serie de platos sucios apilados formando una torre, donde el último plato que se coloca en la pila, el de encima, será el primero en ser lavado, y así sucesivamente hasta llegar al plato que quedó al fondo, el primero en llegar, quien precisamente será el último en ser lavado.

Desarrollo:

Escribir un programa en Java, que utiliza una estructura de tipo pila y que almacenará en ella los valores de números enteros ingresados desde el teclado.

El programa se mostrará en ambiente gráfico, es decir, con un menú gráfico, pantallas, ventanas y botones a gusto del desarrollador. No se permitirá el uso de la consola.

El programa solicita un número entero y con la tecla Enter dicho número se inserta en la pila. El tamaño máximo de la pila será de 50 números enteros.

Cuando se solicite extraer un elemento de la pila, el programa lo mostrará de la siguiente manera:

- Si es un número primo, la ventana donde se muestra el número será de color ROJO.
- Si es un número par, la ventana donde se muestra el número será de color VERDE.
- Si es un número impar (no primo), la ventana donde se muestra el número será de color CELESTE.

Debe crearse un menú con las siguientes opciones:

- Insertar número en la pila.
- Extraer número de la pila.
- Mostrar tamaño de la pila.
- Vaciar pila.

Cada opción debe ser desarrollada en un método separado.

Deben incluirse validaciones para que el programa siempre se ejecute correctamente y sin errores.

Rúbrica:

Aspecto a calificar	Puntos
Inserción en pila	1
Extracción de pila, detectando números primos y mostrando ventana en ROJO.	1
Extracción de pila, detectando números pares y mostrando ventana en VERDE.	1
Extracción de pila, detectando números no primos impares y mostrando ventana en CELESTE.	1
Mostrar tamaño de la pila	1
Vaciar pila	1
Usa un menú gráfico (no en consola)	1
Usa pantallas, ventanas y botones para interactuar con el usuario (el programa no se ejecuta en la consola)	1
El programa se ejecuta sin errores	1
TOTAL	9

Proyecto #1

Objetivo:

Que el estudiante aplique al menos una técnica de ordenamiento utilizando el lenguaje Java.

Explicación:

Para mejorar el procesamiento de cualquier aplicación, se requiere en la mayoría de los casos de algoritmos de ordenamiento de los elementos involucrados en los procesos.

Existen diferentes algoritmos para esto, entre los que están:

- Ordenación por inserción.
- Shellsort.
- Mergesort.
- Quicksort.
- Participación.
- Selección rápida.

El ordenamiento se realiza en memoria siempre y cuando la cantidad de elementos a ordenar no sea tan extenso. Si se trata de un volumen de datos muy amplio, ya se tendría que trabajar sobre disco, bases de datos u otros medios de almacenamiento masivo.

Desarrollo:

Se debe desarrollar un programa en Java, utilizando arreglos y que realice las siguientes tareas:

1. Generar aleatoriamente, una lista de 999 números reales en el rango de 0 a 2000.
2. Ordenar ascendentemente utilizando el método de la Burbuja y mostrar el tiempo empleado.
3. Ordenar ascendentemente utilizando el método Shellsort y mostrar el tiempo empleado.
4. Ordenar ascendentemente utilizando el método Quicksort y mostrar el tiempo empleado.
5. Terminar.

NOTA IMPORTANTE: El programa se debe mostrar en modo gráfico, es decir, con menú y pantallas gráficas, no desde la consola.

Deben incluirse validaciones para que el programa siempre se ejecute correctamente y sin errores.

Rúbrica:

Aspecto a calificar	Puntos
Generación aleatoria de números reales almacenados en arreglos	1
Ordenar ascendentemente utilizando el método de la Burbuja y mostrar el tiempo empleado.	1
Ordenar ascendentemente utilizando el método Shellsort y mostrar el tiempo empleado.	1
Ordenar ascendentemente utilizando el método Quicksort y mostrar el tiempo empleado	1
Usa pantallas en modo gráfico (no usa consola)	1
El programa se ejecuta sin errores	1
TOTAL	6

Proyecto #2

Objetivo:

Que el estudiante aplique sus conocimientos en el uso de listas enlazadas y colas, utilizando el lenguaje java.

Explicación:

Las listas son estructuras simples de datos que se utilizan para almacenar elementos de diferentes tipos. Las listas son ordenadas con longitud arbitraria, donde se pueden añadir o excluir elementos en cualquier ubicación.

Las listas pueden implementarse utilizando arreglos o por medio de listas enlazadas sencillas, donde cada enlace contiene el elemento de la lista así como el puntero al siguiente elemento. Cuando no se tiene claro cuántos elementos se va a requerir almacenar, es cuando se piensa que lo mejor es utilizar listas enlazadas.

Desarrollo:

Se requiere crear un programa en Java que cree una cola utilizando una lista enlazada.

Este programa se utilizará en un cine de nuestra comunidad.

Se requiere crear una lista de películas que se mostrarán en el cine. En esta lista se irá registrando las películas de una en una. Para cada película se debe registrar su nombre, género (seleccionable de una lista que contendrá los elementos: Comedia, Drama, Terror, Infantiles y Acción) y duración de la película. Ésta última se generará aleatoriamente asignándoles un valor entre los 60 y 300 minutos.

De acuerdo a la duración de la película, el cine deberá pagar por los derechos de exposición.

El valor de exposición según la duración de las películas es la siguiente:

DURACIÓN DE LA PELÍCULA	VALOR en \$
60-75	2000
76-90	2500
90-180	3000
181-240	3500
241-300	4000

La lista de películas se deberá manejar como una cola (primero en entrar, primero en salir). La estructura de datos a utilizar serán listas enlazadas. No se debe utilizar ArrayList, LinkedList ni ninguna colección de Java ya implementada, sino implementar listas enlazadas usando nodos.

Se contará con un menú con las siguientes opciones:

1. Insertar película.
2. Extraer película.
3. Obtener el primer elemento de la cola. Se muestra el nombre, género y duración, sin extraerlo.
4. Obtener el último elemento de la cola. Se muestra el nombre, género y duración, sin extraerlo.
5. Obtener la película de la cola en la posición que indique el usuario. Se muestra su nombre, género y duración, sin extraerlo.
6. ¿La cola está vacía?. En esta opción solo se indica SI o NO.
7. Tamaño actual de la cola. Número de películas registradas.
8. Mostrar la lista completa de la cola. Desde el primer elemento hasta el último.
9. Mostrar la inversión total del cine en exposición de las películas registradas.
10. Vaciar cola.
11. Terminar.

Debe utilizarse una interfaz gráfica (menú, pantallas y botones gráficos), es decir, que no sea utilizando la consola.

Deben incluirse validaciones para que el programa siempre se ejecute correctamente y sin errores.

Rúbrica:

Aspecto a calificar	Puntos
Insertar película en la cola	1
Extraer película de la cola	1
Obtener primer elemento de la cola	1
Obtener último elemento de la cola	1
Obtener elemento en determinada posición de la cola según indique el usuario	1
Indicador de cola vacía	1
Tamaño de cola	1
Mostrar cola completa	1
Mostrar inversión total en exposición	1
Vaciar cola	1
Usa un menú gráfica (no desde consola)	1
Uso de pantallas, ventanas y botones gráficos (no desde consola)	1
El programa se ejecuta sin errores	1
TOTAL	13